The Loss Surface of XOR Artificial Neural Networks

Dhagash Mehta,[1,*] Xiaojun Zhao,[1,†] Edgar A. Bernal,[1,‡] and David J. Wales[2,§]

[1] *Systems Department, United Technologies Research Center, East Hartford, CT, USA*
[2] *University Chemical Laboratories, Lensfield Road, Cambridge CB2 1EW, UK*

Training an artificial neural network involves an optimisation process over the landscape defined by the cost (loss) as a function of the the network parameters. We explore these landscapes using optimisation tools developed for potential energy landscapes in molecular science. The number of local minima and transition states (saddle points of index one), as well as the ratio of transition states to minima, grow rapidly with the number of nodes in the network. There is also a strong dependence on the regularisation parameter, with the landscape becoming more convex (fewer minima) as the regularisation term increases. We demonstrate that in our formulation, stationary points for networks with N_h hidden nodes, including the minimal network required to fit the XOR data, are also stationary points for networks with $N_h + 1$ hidden nodes when all the weights involving the additional node are zero. Hence, smaller networks optimized to train the XOR data are embedded in the landscapes of larger networks. Our results clarify certain aspects of the classification and sensitivity (to perturbations in the input data) of minima and saddle points for this system, and may provide insight into dropout and network compression.

I. INTRODUCTION

In recent years, machine learning [1], and particularly deep learning [2, 3], techniques have proved to be highly effective in automating complex tasks. Applications include face and object recognition, scene understanding, natural language processing, speech recognition, game playing, stock-market analysis, and prognostic health management, among others.

A machine learning algorithm can be viewed as a functional mapping between inputs and outputs, the parameters of the mapping being tunable. The learning or training process involves optimisation of the machine parameters to minimize a cost (or loss) function. The loss surface describes the relationship between the values of the parameters and a performance metric, which may include a regularisation term. The reference input-output data points used in the learning stage comprise the training set. The performance metric measures the deviation between the network output and the true output for a given input, and the regularisation term may be included to reduce overfitting.

For some machine learning algorithms, such as linear regression, ridge and lasso regression, *etc.*, the loss surface is usually convex, meaning that there is a single minimum, which simplifies the optimisation task. However, for more sophisticated machine learning techniques, most notably artificial neural networks (ANNs) commonly encountered in deep learning, the loss surface is non-convex [3] with multiple minima. In fact, some studies have demonstrated that the loss surface of single-neuron models may have an exponentially increasing number of minima with increasing number of inputs (see, *e.g.*, [4]). In [5], it is shown that training a network with a single hidden layer with two hidden nodes, one output and n inputs, is a non-deterministic polynomial-time (NP)-complete problem. Commonly used iterative gradient descent-based algorithms for optimisation can converge to local minima rather than the global minimum. Different random initializations of the iterative optimisation process will lead to different local minima, but are unlikely to locate the global minimum reliably unless the number of minima is small.

Despite the large number of local minima expected for highly non-linear and non-convex functions, deep learning frameworks perform reasonably well, as evaluated in terms of performance on previously unseen test data. One of the possible explanations for this observation is that there may be no 'bad' minima at all [6–13], where 'good' and 'bad' minima are loosely defined in terms of the quality of the network performance on the training set (*i.e.*, the empirical error). In [8, 14], however, it was also argued that difficulties in reaching the global minimum in such models arise from the proliferation of saddle points, and further [8], that saddle points give an illusion of a 'bad' minimum because they correspond to higher cost function values. This situation slows down learning, which is typically implemented

* mehtadb@utrc.utc.com
† zhaox@utrc.utc.com
‡ bernalea@utrc.utc.com
§ dw34@cam.ac.uk

using gradient-based methods, such as stochastic gradient descent [15, 16] rather than second-order methods. Alternative methods to escape from saddle points have therefore been proposed, but many of them eventually struggle to deal with the NP-hardness of the problem [17]. The large ratio of saddle points to minima is well known for molecular energy landscapes, both from theory [18] and numerical investigations [19–24].

Recently, [25] showed that the loss surfaces of deep *linear* neural networks, (*i.e.*, multilayer neural networks with activation functions that are linear with respect to the input, but with possibly non-linear loss functions) as well as those of certain special but unrealistic cases of deep nonlinear networks, have degenerate global minima with the same value for the cost function. Moreover, they also showed that for these networks, 'bad' saddle points are indeed present, potentially making the task of reaching a global minimum difficult.

In [26], the loss surface of a fully-connected feed-forward deep network with one output and rectified linear (ReLU) activation functions was approximated by the mean-field spherical p-spin model of statistical physics [27–29], to take advantage of analytical results based on random matrix theory [22, 30–34]. With this approximate model, the authors showed (and later further confirmed [35]) that the number of saddle points with lower indices, at which the value of the loss surface is beyond a certain threshold, diminishes exponentially as the size of the deep network increases. The saddle points and minima at which the value of the loss function is below the threshold were found to be 'good' minima. Hence, the authors concluded that the deeper the network, the less likely it is to find 'bad' minima.

In [36] the ANN landscape was explored systematically using optimisation tools developed in the context of molecular potential energy landscapes [37] (see also [38, 39] for other applications). In particular, a three-layer ANN architecture was employed to fit the Modified National Institute of Standards and Technology (MNIST) data [40], and the resulting landscapes for the cost function were found to exhibit a single-funnel structure [37, 41, 42]. This organization is associated with efficient relaxation to the global minimum in molecular science, and has been identified with 'magic number' clusters, crystallization, and protein-folding landscapes [37, 43, 44]. Although the number of local minima increases exponentially with the number of degrees of freedom, relaxation to the global minimum is effectively guided downhill in energy (the value of the loss function in an ANN landscape) over relatively low barriers [37]. Ref. [36] provides a review, highlighting the connections between molecular potential energy surfaces and ANN landscapes.

In this paper, we present a detailed analysis of the landscape of the loss function for ANNs using tools employed in the analysis of energy landscapes. In particular, we aim to perform a (near) exhaustive search of minima, saddle points of index one (transition states [45]), as well as pathways that connect pairs of minima via transition states. Another goal of this study is to investigate the recently debated quaestion of 'bad' vs 'good' minima in the deep learning literature, in the context of ANNs.

For our purposes the choice of data needs to satisfy certain criteria: (1) the data should be non-trivial, *i.e.*, they should not be linearly separable; (2) the architecture of the ANN to completely fit the chosen data should be known; (3) the ANN to completely fit the data should only have a small number of degrees of freedom, so that we can obtain an essentially complete set of minima. We considered various public datasets, including the well-known MNIST database of hand-written digits. However, these did not satisfy our criteria for the present investigation. Moreover, the current best network for the MNIST dataset has far too many degrees of freedom (the number of network parameters) for an exaustive search of local minima to be practical. In contrast, the exclusive OR (XOR) data is both non-trivial (XOR is non-linearly separable) and simple enough to satisfy our criteria. Furthermore, the loss function of various ANNs that fit the XOR function has been studied in many previous contributions, as reviewed in the next section. Hence, we focus on the XOR function in this paper.

Our principal results are: (1) a complete characterization of local minima (*i.e.*, all the different possible trainings) as well as transition states and pathways between them; (2) a classification of 'good' and 'bad' minima based on the sensitivity of the corresponding trained network to perturbation of the input data, as well as the sparsity (§II) of the trained network; (3) identification of 'bad' minima, which exist even for the simple XOR function; (4) demonstration that as long as an ANN with one hidden layer is overspecified (*i.e.*, the hidden layer has more neurons than the minimum number required to fit the data), the landscape contains an optimal ANN as a minimum of the loss function where all the incoming and outgoing weights for certain neurons vanish.

In the remaining sections we describe the XOR function, explain the set up of the problem, and provide a brief summary of the relevant results.

II. LOSS FUNCTION OF XOR

The exclusive OR (XOR) between two boolean variables is a logical operation whose output is true only when the two inputs have different values. This function has been the subject of several studies aimed at gaining insight into properties of loss functions for small neural networks, because of the nonlinear separability of the data [46–51].

The neural network in our study has one input layer with two nodes, one hidden layer with N_h nodes, and one output layer with two nodes. The training set comprises two inputs with four possible combinations $\mathbf{X} = \{(0,0),(0,1),(1,0),(1,1)\}$ and their corresponding outputs $\mathbf{Y} = \{0,1,1,0\}$. The two output nodes correspond to probabilities assigned by the network to possible output values 0 and 1. For each node j in the hidden layer, a bias ω_j^{bh} is added to the sum of the corresponding weights used in the activation function. Similarly, for each node i in the output layer, a bias ω_i^{bo} is added to the sum of corresponding weights. Hence, with the weights on the links between hidden node j and output i being $\omega_{ij}^{(1)}$ and those on the links between input k and hidden node j being $\omega_{jk}^{(2)}$, and with tanh as the activation function, the input-output relationship of the network in question is

$$y_i = \omega_i^{\mathrm{bo}} + \sum_{j=1}^{N_h} \omega_{ij}^{(1)} \tanh\left(\omega_j^{\mathrm{bh}} + \sum_{k=1}^{2} \omega_{jk}^{(2)} x_k\right), \quad (1)$$

where the internal sum from 1 to 2 corresponds to the two components of the input data. The outputs are then converted into softmax probabilities using

$$p_c(\mathbf{W};\mathbf{X}) = e^{y_c} / \left(e^{y_0} + e^{y_1}\right). \quad (2)$$

To train the model, we minimize

$$E(\mathbf{W};\mathbf{X}) = -\frac{1}{|\mathbf{X}|} \sum_{d=1}^{|\mathbf{X}|} \ln p_{c(d)}(\mathbf{W};\mathbf{X}) + \lambda \mathbf{V}^2, \quad (3)$$

with respect to $\mathbf{W} = \{\omega_{ij}^{(1)}, \omega_{jk}^{(2)}, \omega_j^{\mathrm{bh}}, \omega_i^{\mathrm{bo}}\}$, where $c(d)$ is the correct outcome for data item d. The loss function includes a performance and a regularisation term; $\mathbf{V}$ is either a vector containing the $\omega_{i,j}^{(1)}$ and $\omega_{j,k}^{(1)}$ parameters, or alternatively all the parameters $\mathbf{V} = \mathbf{W}$. We will return to the specific choice of $\mathbf{V}$ below. $|\mathbf{X}| = 4$ is the number of data points, which corresponds to the cardinality of the training input set, and λ is the regularisation parameter. In our experiments, we consider $\lambda = 10^{-l}$ for $l = 1, 2, \ldots, 6$. In Eq. (3), cross-entropy is used as the performance term, where one of the probabilities in the true distribution is one and the others are zero. Hence the summation reduces to a single term $-\ln p_{c(d)}$ for each data item.

Depending on the formulation, the XOR loss function may possess a range of discrete and continuous symmetries [52–54]. Hence, once a minimum or a stationary point is found, others can be found having the same loss function value (infinitely many if continuous symmetries exist, corresponding to zero eigenvalues of the Hessian second derivative matrix). In the present work these degenerate solutions are recognised by tight convergence of the loss function, and are lumped together.

For the loss function defined in Eq. (3) it is straightforward to show that a stationary point for a network with N_h hidden nodes is also a stationary point of some index for a network with $N_h + 1$ hidden nodes if all the weights involving the additional node are zero (see Appendix A). The extra degrees of freedom associated with the larger network introduce more flexibility, which we would expect to lead to a lower value for the loss function after relaxation, and perhaps to a higher Hessian index. However, the regularisation term introduces a competing effect, so the stationary point corresponding to the augmented network with zero weights for hidden node $N_h + 1$ could be a minimum or a saddle point. Larger values of λ will penalise non-zero values for the additional weights, and are therefore more likely to conserve the Hessian index. We refer to the network in which at least one of the weights is zero as a sparse network; for a given network, the larger the number of zero weights, the sparser the network.

A. Previous Analysis of the XOR Loss Landscape

It was initially shown that one particular formulation of the loss function for the simplest network required to solve the XOR problem has only one minimum [46–48]. Ref. [50] demonstrated the absence of higher local minima in more complex networks (networks with two hidden layers and two units in each layer) as long as the activation units are not saturated [49]; in contrast, when the activation units saturate due to some weights having effectively infinite values, local minima start to appear. The existence of suboptimal local minima in landscapes of more complicated networks (networks with two hidden layers, two units in the first layer and three units in the second layer) was demonstrated in [51], di-

rectly contradicting earlier assertions that two-layer neural networks with sigmoid activation functions and $N_h - 1$ hidden nodes do not have suboptimal local minima when the learning is performed with N_h training samples [55]. Ref. [14] reported multiple minima for the XOR problem for specific neural network architectures. Recently, the authors of [56] took a bottom-up approach in which they analyzed the loss function of a simple ('flattened') neural network, which aims to approximate the XOR function, and found 'bad' minima where learning fails. A thorough review of the studies of optimisation landscape of the ANNs for XOR up to 2001 can be found in [57] (see also [58]). We emphasise that the loss functions in previous work are often constructed differently, so the resulting landscapes may not be directly comparable.

Our inclusion of regularisation over all degrees of freedom in the present work is a key difference from some previous studies, and simplifies the characterisation of stationary points. Without sufficient regularisation, the machine learning landscape is likely to include very flat regions, probably including non-Morse stationary points with zero Hessian eigenvalues [48]. We explicitly wish to exclude such possibilities in the present analysis.

III. ENERGY LANDSCAPE THEORY AND COMPUTATIONAL METHODS

In chemical physics, the hypersurface defined by the potential energy function, a multivariate nonlinear function of $3N$ atomic coordinates for N atoms, is referred to as the potential energy landscape [37]. The most interesting points on the energy landscape are usually stationary points where the gradient vanishes. Stationary points are further classified according to the number of negative eigenvalues, or the index i, of the Hessian (second derivative) matrix. Stationary points of index $i = 0$ are minima, where any small displacement of internal degrees of freedom raises the energy. Local minima are connected by geometrically defined steepest-descent paths from transition states, which are saddles of index one [45]. Non-Morse stationary points have zero eigenvalues that do not result from continuous symmetries of the Hamiltonian.

The computational methods employed for geometry optimisation and construction of connected networks of local minima and transition states are well established, and a brief summary is provided here. More details are available in reviews [37], including a recent contribution that focuses on machine learning landscapes [36].

Global minima and a survey of low-lying local minima were obtained by basin-hopping global optimisation [59–61], using our `GMIN` program [62]. Here, steps are taken between local minima, obtained via random changes to the coordinates of the current minimum in the chain (in our case, the network weights), with an acceptance criterion based on the change in the loss function scaled by a fictitious temperature parameter.

To determine connections between local minima via transition states we first run double-ended searches between specific pairs using the doubly-nudged [63] elastic band [64, 65] approach, which interpolates between the end points via a sequence of images. The images corresponding to local maxima are then converged to transition states using the single-ended hybrid eigenvector-following algorithm [66–69]. For each transition state, the two connected minima are determined by calculating (approximately) the two geometrically defined steepest-descent paths. The OPTIM program [70] was used for all these calculations.

Although this methodology is well established, some additional effort was required to tighten convergence criteria and ensure the accuracy of the pathways in terms of the connectivity. These changes were necessary because the landscapes in question support very soft degrees of freedom, even when regularisation is included over all variables. For example, with a single hidden node, the minimum with zero weights is connected via a transition state with weights that are very small in magnitude, and the difference in the loss function is only 0.1425×10^{-10}. All transition states were therefore checked using eigenvector-following with analytical second derivatives, and the steepest-descent paths were obtained using a second-order method [71] after determining the displacement from the transition state that maximised the decrease in loss function. This procedure produced consistent connections for all the transition states obtained when they were located in alternative runs.

Additional checks were performed to ensure that stationary points with zero Hessian eigenvalues were not included in the databases. Here, zero eigenvalues were first defined using a cutoff of 10^{-9}, which is about an order of magnitude less than the values observed for the smallest legitimate eigenvalues. Changing the cutoff to 10^{-10} for $\lambda = 10^{-6}$ and rerunning connection attempts between all pairs of minima did not produce any additional stationary points.

When regularisation is applied to all variables, there are no zero Hessian eigenvalues caused by continuous symmetries, but singularities arise if the bias weights are not regularised. These degrees of freedom could be treated by projection and eigenvalue shifting [37] using analytical expressions for the corresponding Hessian eigenvectors. In molecular geometry optimisation, continuous symmetries arise from overall translation and rotation [37]; in neural networks, a uniform displacement in the w_i^{bo} has no effect on the probabilities. The additional zero Hessian eigenvalues pose no problems for the custom LBFGS minimisation routine (a limited memory version of the quasi-Newton Broyden [72], Fletcher [73], Goldfarb [74], Shanno [75] BFGS procedure) employed in `GMIN` [62]. However, shifting and projection would be required to locate transition states and construct disconnectivity graphs (see §III A). We therefore restrict our landscape characterisations to loss functions with regularisation applied to all the variables, where the connectivity is well defined.

The database of minima and transition states resulting from systematic connection attempts between pairs of local minima constitutes a transition network [76–79]. Various techniques have been described for refining such networks [80–82] (See [36, 83]. In the present work we employed the PATHSAMPLE program [84] to distribute OPTIM jobs and organize the resulting output to expand the stationary point databases. The overall approach is known as discrete path sampling [85, 86].

A. Disconnectivity Graphs

The energy landscape is a high-dimensional object, which usually cannot be visualized effectively as a three-dimensional surface. Instead, disconnectivity graphs [43, 87–89] provide a powerful approach for understanding the organization, faithfully representing the barriers between local minima. The vertical axis in a disconnectivity graph corresponds to the energy (cost or loss function), and each branch terminates at the value for a local minimum. At a regular series of threshold values, the local minima are grouped into disjoint superbasins [87], whose members can interconvert via transition states without exceeding the threshold. Branches join together at the threshold where they can interconvert. The horizontal axis is usually chosen so that the branches are spaced out and do not overlap; order parameters can also be employed to arrange the branches [90], or to color them.

IV. RESULTS AND DISCUSSION

In this section, we present our results and discuss their interpretation. In particular, we consider the dependence between N_h and λ and the nature of the resulting landscape, the relationship between the complexity of the network and energy values, and introduce an empirical analysis of network sensitivity to perturbation of the inputs.

A. Machine Learning Landscapes

Machine learning landscapes were constructed for the loss function of Eq. (3) with various combinations of regularisation parameters and hidden nodes. Selected examples are illustrated here using disconnectivity graphs [43, 87–89]. Fig. 1 shows the landscapes obtained for $\lambda = 10^{-6}$ with one to six hidden nodes, and Figs 2–6 show how the landscape changes with λ for six hidden nodes. In these graphs, the terminus of each branch corresponds to the value of the minimised loss function, with the global minimum at the bottom. Solutions with identical values are lumped together, so the branches that appear degenerate in Fig. 1 (a) and (b) actually correspond to slightly different loss values. For three hidden nodes and above, networks corresponding to all the local minima represented in Fig. 1 provide an accurate fit of the four input data points. To quantify the prediction quality, we calculated the area under the curve (AUC) values for receiver operating characteristic (ROC) plots. All the AUC values obtained for the training set of four possible input values are close to unity, except for the coloured branches in Fig. 1 (a) and (b). These results show that once sufficient network parameters are included, increasing the number of hidden nodes results in additional solutions corresponding to local minima with only slightly different loss function values.

For any values of N_h and λ, the trivial solution where all weights are zero is always a bad minimum (as defined below). In some cases including this minimum in the disconnectivity graphs changes the scale dramatically. Hence, we omit this minimum in most of the graphs. Note that in [91], it was shown that if the network architecture consists of parallel subnetworks where each subnetwork has a particular architecture defined by a specific elemental mapping, a minimum at which all weights in one of the subnetworks are zero is the global minimum. However, this formulation is different from the present work.

Figs. 2–6 illustrate the effect of the regularisation

parameter when applied over all the variables, including the bias weights. λ basically sets the vertical scale, corresponding to the optimised loss function value, and determines how many minima the machine learning landscape can support. All the local minima in these graphs correspond to AUC values close to unity. The predicted probabilities for the correct output corresponding to the two minima in Fig. 2 vary between 0.85 and 0.91. For minima in Figs. 3–6 corresponding to $\lambda = 10^{-3}$ and smaller, the probabilities are 0.99 or better in each case.

The variation in the number of minima and transition states as a function of λ for a fixed network architecture can be understood from catastrophe theory. As λ increases from zero, minima and transition states merge via fold catastrophes, and the remaining uphill barriers from lower to higher energy minima generally increase. A detailed analysis for surfaces parameterized by a single parameter (such as λ in our case) is provided in [92, 93]. The decreasing number of minima (*i.e.*, increasing convexity) observed when varying λ between 10^{-6} and 10^{-2} is also known as topology trivialization [31] in statistical physics, a phenomenon that has been noted for various energy surfaces [30, 94–96] including ANNs [97].

B. Visualizing Networks at Minima

In the insets of Figs. 2-6, we provide visualizations of the ANN at a few representative minima. We represent each minimum as a network: if a weight *at the minimum* is numerically zero (*i.e.*, 10^{-10} or smaller), we do not include a connection between the corresponding nodes. The figures show that for $\lambda = 10^{-2}$, for one of the two minima the network is fully connected, whereas for the other minimum exactly three neurons are connected and the other three are disconnected. For other values of λ, we find some minima at which zero, one, or two neurons are disconnected, whereas all minima for $\lambda = 10^{-6}$ correspond to fully connected networks.

In the absence of separate training and testing data, the quality of the solutions corresponding to local minima in the learning process of a network has usually been measured in terms of the network performance on the training set at the weight values determined by the minima in question, *i.e.*, the empirical error. Here, we quantify the quality of minima by considering both the associated empirical error and the capacity of the resulting network, or Vapnik-Chervonenkis (VC) dimension [98], which can be intuitively interpreted as the number of tunable parameters in a neural network. According to the structural risk minimization principle, first proposed by Vapnik [99], the optimal minimum corresponds to the model with the smallest combined empirical risk and capacity. While increasing model complexity is usually accompanied by decreased empirical error, higher capacity can also lead to overfitting, or the inability to generalize beyond the training set. We restrict our analysis to models with enough capacity to fit the training data, and note that models with less complexity than required will be penalized by the performance term in the structural risk expression.

In our calculations, the performance metric is the AUC, and the complexity of the model is coarsely measured as the number of non-zero weights in the network. In the XOR case, we find two separate regions of λ (at the discrete values we have chosen): for $\lambda = 10^{-1}$ and $\lambda = 10^{-2}$, there exists at least one 'bad' minimum (in terms of empirical error), whereas for $\lambda = 10^{-3}, \ldots, 10^{-6}$ no such solutions are found except for the trivial minimum with zero weights. However, for some but not all of the 'good' minima, many of the links have a number of weights that are practically zero, which indicates that less complex models suffice to successfully classify the four data points in the training set. There is only one (up to the discrete symmetries) minimum at $\lambda = 10^{-2}$ in which only as many weights are non-vanishing as are needed to construct a minimal neural network to fit the XOR function, *i.e.*, a fully connected network with three hidden nodes. Hence, if we define a suboptimal minimum as one in which more than the necessary number of weights have non-vanishing values (*i.e.*, one at which more than the necessary number of neurons remain connected), then there is only one optimal minimum: the other minima are suboptimal.

C. Optimal Network Configuration

The minimal network we obtained in our experiments, for any value of $N_h > 2$, has three fully connected hidden nodes (*i.e.*, all incoming and outgoing weights are non-zero for these three neurons). However, it is well-known that the XOR data can be fitted with a network having $N_h = 2$. Hence a configuration with only non-zero weights that connect exactly two neurons to the inputs and outputs, in addition to the bias weights, should be the 'best' configuration for networks with $N_h > 2$. In fact, this configuration is not a minimum but is always a saddle point of index $N_h - 2$ for the values of λ consid-

ered here. In other words, for the XOR data with the present network set up, when the size of the network is larger than the minimal network, the minimal network configuration is a saddle point. This conclusion differs from a recent study [100] which showed that deep networks converge to saddle points at which the Hessian matrix is singular, because in our case the 'best' saddle point is not degenerate. Details of this computation will be discussed elsewhere.

D. Network Sensitivity to Perturbations in the Input Data

Neural networks are susceptible to so-called adversarial examples, where small perturbations in the input can cause misclassification [101]. For networks operating on high-dimensional spaces, effective adversarial perturbations need to be carefully engineered. In the present work the effect of perturbations can be analysed in detail given the low dimensionality of the input space. The sensitivity of a trained network to perturbations in the input data is also referred to as *stability* of the network [102] (see, *e.g.*, [103], for a recent attempt to relate generalization and geometry of landscapes of ANNs). To this end, the output of the network to inputs in the range $[-0.5, 1.5]$ with a step size of 0.015 was computed. The insets of Figs. 2-6 located below the network visualization of each of the selected minima illustrate the results. The color of the data point with (x, y) coordinates corresponds to the output of the network for input values of x and y: red and blue points correspond to 1 and 0 outputs, respectively. The white triangle and square symbols represent the 0 and 1 output, respectively, for the actual inputs present in the training set. Intuitively, a stable network should output 0 when the inputs are similar to each other and 1 otherwise, regardless of the actual values. Specifically, and given the choice of the binary coding scheme $(x, y) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$, for inputs satisfying $|x - y| \leq 0.5$, the output of the network should be 0, with a desired output of 1 for every other input combination. Inspection of Figs. 2-6 reveals that the sparser networks are more robust, although the converse is not always true.

V. CONCLUSION AND OUTLOOK

Using the energy landscape theory developed in chemical physics, we have investigated the optimisation landscape of the loss function for neural networks trained to approximate the XOR function. Our network has one hidden layer with N_h neurons in the hidden layer. We find that the number of minima and saddle points of index 1 change rapidly with N_h and regularisation parameter λ. More importantly, we discovered that the loss surface includes minima where some of the weights are essentially zero (around $\sim 10^{-10}$ or below) such that some of the hidden neurons effectively appear disconnected. The number of disconnected neurons can vary with λ. In particular, for the XOR data and for any $N_h > 3$, there is always a minimum at which exactly three hidden neurons remain connected, with all the remaining $N_h - 3$ neurons being disconnected, indicating that a network with 3 hidden neurons is a minimal configuration to successfully separate the XOR data, whereas a 2-neuron configuration (which can also separate the data) is found to be a saddle point of index $N_h - 2$ rather than a minimum.

The universal approximation theorem states that a feed-forward ANN with one hidden layer and finite N_h can approximate any continuous function defined on compact subsets of the real space. The theorem assumes mild criteria on the activation function. All these criteria are satisfied by the ANN and the activation function (hyperbolic tangent) we have chosen in the present work. The theorem, however, does not yield a procedure to obtain the optimal number N_h to approximate the function. At the very least, determining the optimal number would require a priori knowledge of the data, for example, how much data is available, whether the data includes all the representative cases of the complete dataset, the amount of noise, *etc.* Our study does not address the question of how to find the optimal number of hidden neurons analytically. However, we have shown empirically for the XOR example that if we select a network with a larger number of neurons than the optimal number required to fit the data, then for a certain value of the regularization parameter there will be a minimum in the landscape at which the corresponding network will have only the optimal number of neurons connected and all the others effectively disconnected.

Our results may also explain why a large number of network parameters are found redundant in previous studies [104, 105]: if a model with more parameters than needed to fit the data is used, a minimum with sufficient regularisation will make the unwanted network parameters redundant during minimization. Our approach in turn provides a systematic way to compress networks, a topic that has recently attracted significant attention for fast and low-power mobile applications [106, 107] (see also ref. [108] for an earlier attempt to use an optimization approach

to reduce the number of network parameters, and [109] for a review on other methods to prune ANNs).

We note that linear regression with L_2-regularisation guarantees a unique minimum at which the unimportant features are removed from the model, whereas the landscape of ANNs with L_2-regularisation consists of multiple minima with differing numbers of zero weights. Our results indicate that scanning the landscape may provide a systematic way to perform partial hyperparameter parameterization of ANNs, meaning optimisation of the number of hidden layers and the number of neurons in each layer. Verifying this result for more complex datasets as well as for deeper ANNs may shed further light on the the more complex nature of the learning process. An effort to investigate this issue is in progress.

Recent empirical findings on regularisation techniques aimed at decreasing coadaptation across different neurons (*e.g.*, dropout [110] and spatial dropout [111]) indicate that one of the reasons complex models have a tendency to overfit the training data is that, on training, some units may learn to correct mistakes made by other units. This compensation may be effective at improving performance on the training set, but usually leads to poor generalization capabilities. Our results reinforce this hypothesis, since simpler models showcase smaller risk for coadaptation, as there are fewer potential symbiotic relationships among neurons available at training.

Finally, we empirically demonstrated that sparse networks tend to exhibit improved (*i.e.*,reduced) sensitivity to perturbations in the input, as evidenced by the 0-valued responses around the diagonal defined by $x = y$. In contrast, for other minima, undesired regions of 1-valued responses occur around the $y = 1 - x$ diagonal. This trend was observed across a range of values for the regularisation parameter λ. This analysis quantitatively distinguishes between good and bad training: if the learning is not done carefully, then it could lead to a network corresponding to a minimum in the loss function that may appear good enough for the specific data, but could be sensitive to perturbations in the inputs. Such a system may be more vulnerable to adversarial attacks. Whether such minima survive when more hidden layers are added is an open issue and should be further investigated.

In the future, we plan to devise an algorithm that directly finds the best minimum in the minimal network sense (and the corresponding value of the regularisation parameter), and extend these investigations to deeper networks and larger datasets, which may help to resolve potentially more complex landscapes and issues concerning zero eigenvalues of the Hessian for the cost function, which result in 'flat' minima [112, 113].

ACKNOWLEDGEMENT

DM, XZ and EAB acknowledge internal funding from UTRC. DJW acknowledges financial support from the EPSRC. We thank Jose-Miguel Pasini, Kishore Reddy, Kunal Srivastava and Amit Surana for their feedback.

[1] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[2] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[3] Y. Bengio, I. J. Goodfellow, and A. Courville. Deep learning. *MIT Press.*, 2015.

[4] P. Auer, M. Herbster, and M. K. Warmuth. Exponentially many local minima for single neurons. *Adv. in Neural Info. Proc. Sys.*, pages 316–322, 1996.

[5] A. Blum and R. L. Rivest. Training a 3-node neural network is np-complete. In *Proceedings of the 1st International Conference on Neural Information Processing Systems*, pages 494–501. MIT Press, 1988.

[6] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2:53–58, 1989.

[7] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[8] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

[9] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017.

[10] I. Safran and O. Shamir. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning*, pages 774–782, 2016.

[11] I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

[12] H. W. Lin and M. Tegmark. Why does deep and cheap learning work so well? *arXiv preprint arXiv:1608.08225*, 2016.

[13] D. Soudry and Y. Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

[14] F. M. Coetzee and V. L. Stonick. 488 solutions to the xor problem. *Advances in Neural Information Processing Systems*, pages 410–416, 1997.

[15] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:142, 1998.

[16] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[17] A. Anandkumar and R. Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on Learning Theory*, pages 81–102, 2016.

[18] D. J. Wales and J. P. K. Doye. Stationary points and dynamics in high-dimensional systems. *J. Chem. Phys.*, 119:12409–12416, December 2003.

[19] J. P. K. Doye and D. J. Wales. Saddle points and dynamics of Lennard-Jones clusters, solids, and supercooled liquids. *Journal Chem. Phys.*, 116:3777–3788, 2002.

[20] D. Mehta and M. Kastner. Stationary point analysis of the one-dimensional lattice landau gauge fixing functional, aka random phase xy hamiltonian. *Annals of Physics*, In Press:–, 2011.

[21] D. Mehta. Finding All the Stationary Points of a Potential Energy Landscape via Numerical Polynomial Homotopy Continuation Method. *Phys. Rev.*, E84:025702, 2011.

[22] D. Mehta, D. A. Stariolo, and M. Kastner. Energy landscape of the finite-size spherical three-spin glass model. *Phys. Rev.*, E87(5):052143, 2013.

[23] Dhagash Mehta, Matthew Niemerg, and Chuang Sun. Statistics of stationary points of random finite polynomial potentials. *Journal of Statistical Mechanics: Theory and Experiment*, 2015:P09012, 2015.

[24] D. Mehta, C. Hughes, M. Kastner, and D. J. Wales. Potential energy landscape of the two-dimensional xy model: Higher-index stationary points. *The Journal of chemical physics*, 140:224503, 2014.

[25] K. Kawaguchi. Deep learning without poor local minima. *arXiv preprint arXiv:1605.07110*, 2016.

[26] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. *arXiv preprint arXiv:1412.0233*, 2014.

[27] A. Crisanti and H. J. Sommers. The spherical p-spin interaction spin-glass model - the statics. *Z. Fur Physik B-condensed Matter*, 87:341–354, 1992.

[28] J Kurchan, G Parisi, and Miguel Angel Virasoro. Barriers and metastable states as saddle points in the replica approach. *Journal de Physique I*, 3:1819–1838, 1993.

[29] T. Castellani and A. Cavagna. Spin-glass theory for pedestrians. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:P05012, 2005.

[30] Y. V. Fyodorov and P. Le Doussal. Topology trivialization and large deviations for the minimum in the simplest random optimization. *J. of Stat. Phys.*, 154:466–490, 2014.

[31] Y. V. Fyodorov. High-dimensional random fields and random matrix theory. *arXiv preprint arXiv:1307.2379*, 2013.

[32] A. Auffinger, G. B. Arous, and J. Černỳ. Random matrices and complexity of spin glasses. *Communications on Pure and Applied Mathematics*, 66:165–201, 2013.

[33] A. Auffinger and G. B. Arous. Complexity of random smooth functions on the high-dimensional sphere. *The Annals of Probability*, 41:4214–4247, 2013.

[34] D. Mehta, J. D. Hauenstein, Matthew Niemerg, Nicholas J. Simm, and Daniel A. Stariolo. Energy Landscape of the Finite-Size Mean-field 2-Spin Spherical Model and Topology Trivialization. 2014.

[35] L. Sagun, V. U. Guney, G. B. Arous, and Y. LeCun. Explorations on high dimensional landscapes. *arXiv preprint arXiv:1412.6615*, 2014.

[36] A. J. Ballard, S. Martiniani, D. Mehta, J. D. Stevenson, and D. J. Wales. Energy landscapes for machine learning. *Phys. Chem. Chem. Phys.*, 19:12585–12603, 2017.

[37] D. J. Wales. *Energy Landscapes*. Cambridge University Press, Cambridge, 2003.

[38] M. Pavlovskaia, K. Tu, and S.-C. Zhu. Mapping the energy landscape of non-convex optimization problems. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 8932, pages 421–435, 2015.

[39] A. J. Ballard, J. D. Stevenson, R. Das, and D. J. Wales. Energy landscapes for a machine learning application to series data. *The Journal of chemical physics*, 144:124119, 2016.

[40] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[41] J. P. K. Doye, D. J. Wales, and M. A. Miller. Thermodynamics and the global optimization of lennard-jones clusters. *J. Chem. Phys.*, 109:8143–8153, 1998.

[42] J. P. K. Doye, M. A. Miller, and D. J. Wales. The double-funnel energy landscape of the 38-atom lennard-jones cluster. *J. Chem. Phys.*, 110:6896–6906, 1999.

[43] D. J. Wales, M. A. Miller, and T. R Walsh. Archetypal energy landscapes. *Nature*, 394:758–760, 1998.

[44] D. J. Wales and T. V. Bogdan. Potential energy and free energy landscapes. *J. Phys. Chem. B*, 110:20765–20776, 2006.

[45] J. N. Murrell and K. J. Laidler. Symmetries of activated complexes. *Trans. Faraday. Soc.*, 64:371–377, 1968.

[46] I. G. Sprinkhuizen-Kuyper and E. J. W. Boers. The error surface of the simplest xor network has no local minima, 1994.

[47] I. G. Sprinkhuizen-Kuyper and Egbert J. W. Boers. The error surface of the simplest xor network has only global minima. *Neural Comput.*, 8:1301–1320, August 1996.

[48] L. G. C. Hamey. Xor has no local minima: A case study in neural network error surface analysis. *Neural Netw.*, 11:669–681, June 1998.

[49] I. G. Sprinkhuizen-Kuyper and Egbert J. W. Boers. The error surface of the 2-2-1 xor network: The finite stationary points. *Neural networks*, 11(4):683–690, June 1998.

[50] I. G. Sprinkhuizen-Kuyper and E. J. W. Boers. The local minima of the error surface of the 2-2-1 xor network. *Annals of Mathematics and Artificial Intelligence*, 25:107, 1999.

[51] I. G. Sprinkhuizen-Kuyper and E. J. W. Boers. A local minimum for the 2-3-1 xor network. *IEEE Transactions on Neural Networks*, 10:968–971, Jul 1999.

[52] H. J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural networks*, 5:589–593, 1992.

[53] A. M. Chen, H.-M. Lu, and R. Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural computation*, 5:910–927, 1993.

[54] V. Badrinarayanan, B. Mishra, and R. Cipolla. Symmetry-invariant optimization in deep networks. *arXiv preprint arXiv:1511.01754*, 2015.

[55] X. H. Yu. Can backpropagation error surface not have local minima. *IEEE Transactions on Neural Networks*, 3:1019–1021, Nov 1992.

[56] G. Swirszcz, W. M. Czarnecki, and R. Pascanu. Local minima in training of neural networks. *stat*, 1050:17, 2017.

[57] M. Gallagher. *Multi-layer perceptron error surfaces: visualization, structure and modelling.* PhD thesis, The University of Queensland, Australia., 2000.

[58] R. Rojas. *Neural networks: a systematic introduction.* Springer-Verlag New York, Inc., 1996.

[59] Z. Li and H. A. Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA*, 84:6611–6615, 1987.

[60] D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101:5111–5116, 1997.

[61] D. J. Wales and H. A. Scheraga. Global optimization of clusters, crystals and biomolecules. *Science*, 285:1368–1372, 1999.

[62] D. J. Wales. Gmin: A program for basin-hopping global optimisation, basin-sampling, and parallel tempering. http://www-wales.ch.cam.ac.uk/software.html.

[63] S. A. Trygubenko and D. J. Wales. A doubly nudged elastic band method for finding transition states. *J. Chem. Phys.*, 120:2082–2094, 2004.

[64] G. Henkelman and H. Jónsson. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *J. Chem. Phys.*, 113:9978–9985, 2000.

[65] G. Henkelman, B. P. Uberuaga, and H. Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.*, 113:9901–9904, 2000.

[66] L. J. Munro and D. J. Wales. Defect migration in crystalline silicon. *Phys. Rev. B*, 59:3969–3980, 1999.

[67] G. Henkelman and H. Jónsson. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *J. Chem. Phys.*, 111:7010–7022, 1999.

[68] Y. Kumeda, L. J. Munro, and D. J. Wales. Transition states and rearrangement mechanisms from hybrid eigenvector-following and density functional theory. application to c10h10 and defect migration in crystalline silicon. *Chem. Phys. Lett.*, 341:185–194, 2001.

[69] Y. Zeng, P. Xiao, and G. Henkelman. Unification of algorithms for minimum mode optimization. *J. Chem. Phys.*, 140:044115, 2014.

[70] D. J. Wales. Optim: A program for geometry optimisation and pathway calculations. http://www-wales.ch.cam.ac.uk/software.html.

[71] M. Page and J. W. McIver. ? *J. Chem. Phys.*, 88:922, 1988.

[72] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *J. Inst. Math. Appl.*, 6:76–90, 1970.

[73] R. Fletcher. A new approach to variable metric algorithms. *Comput. J.*, 13:317–322, 1970.

[74] D. Goldfarb. A family of variable-metric methods derived by variational means. *Math. Comput.*, 24:23–26, 1970.

[75] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Math. Comput.*, 24:647–656, 1970.

[76] F. Rao and A. Caflisch. The protein folding network. *J. Mol. Biol.*, 342:299–306, 2004.

[77] F. Noé and S. Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Op. Struct. Biol.*, 18:154–162, 2008.

[78] D. Prada-Gracia, J. Gómez-Gardenes, P. Echenique, and F. Fernando. Exploring the free energy landscape: From dynamics to networks and back. *PLoS Comput. Biol.*, 5:1–9, 2009.

[79] D. J. Wales. Energy landscapes: Some new horizons. *Curr. Op. Struct. Biol.*, 20:3–10, 2010.

[80] J. M. Carr, S. A. Trygubenko, and D. J. Wales. Finding pathways between distant local minima (7 pages). *J. Chem. Phys.*, 122:234903, 2005.

[81] B. Strodel, C. S. Whittleston, and D. J. Wales. Thermodynamics and kinetics of aggregation for the gnnqqny peptide. *J. Amer. Chem. Soc.*, 129:16005–16014, 2007.

[82] D. J. Wales, J. M. Carr, M. Khalili, V. K. de Souza, B. Strodel, and C. S. Whittleston. Pathways and rates for structural transformations of peptides and proteins. In *Proteins: Energy, Heat and Signal Flow*, Computation in chemistry, page 315. CRC Press, 2009.

[83] J. A. Joseph, K. Roder, D. Chakraborty, R. G. Mantell, and D. J. Wales. Exploring biomolecular energy landscapes. *Chem. Commun.*, 2017.

[84] D. J. Wales. Pathsample: A program for generating connected stationary point databases and extracting global kinetics. http://www-wales.ch.cam.ac.uk/software.html.

[85] D. J. Wales. Discrete path sampling. *Mol. Phys.*, 100:3285–3306, 2002.

[86] D. J. Wales. Some further applications of discrete path sampling to cluster isomerization. *Mol. Phys.*, 102:891–908, 2004.

[87] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *J. Chem. Phys.*, 106:1495, 1997.

[88] S. V. Krivov and M. Karplus. Free energy disconnectivity graphs: Application to peptide models. *J. Chem. Phys.*, 117:10894–10903, 2002.

[89] D. A. Evans and D. J. Wales. Free energy landscapes of model peptides and proteins. *J. Chem. Phys.*, 118:3891–3897, 2003.

[90] T. Komatsuzaki, K. Hoshino, Y. Matsunaga, G. J. Rylance, R. L. Johnston, and D. J. Wales. How many dimensions are required to approximate the potential energy landscape of a model protein? (9 pages). *J. Chem. Phys.*, 122:084714, 2005.

[91] B. D. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

[92] D. J. Wales. A microscopic basis for the global appearance of energy landscapes. *Science*, 293:2067–2070, 2001.

[93] T. V. Bogdan and D. J. Wales. New results for phase transitions from catastrophe theory. *J. Chem. Phys.*, 120:11090–11099, 2004.

[94] M. Kastner and D. Mehta. Phase transitions detached from stationary points of the energy landscape. *Phys. Rev. Lett.*, 107:160602, 2011.

[95] D. Mehta, J. D. Hauenstein, and M. Kastner. Energy-landscape analysis of the two-dimensional nearest-neighbor φ 4 model. *Phys. Rev. E*, 85:061103, 2012.

[96] D. Mehta, D. A. Stariolo, and M. Kastner. Energy landscape of the finite-size spherical three-spin glass model. *Phys. Rev. E*, 87:052143, 2013.

[97] P. Chaudhari and S. Soatto. Trivializing the energy landscape of deep networks. *arXiv preprint arXiv:1511.06485*, 2015.

[98] V. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16:264–280, 1971.

[99] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[100] A. R. Sankar and V. N. Balasubramanian. Are saddles good enough for deep learning? *arXiv preprint arXiv:1706.02052*, 2017.

[101] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[102] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[103] H. Li, Z. Xu, G. Taylor, and T. Goldstein. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

[104] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, and M. Ranzato. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pages 2148–2156, 2013.

[105] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, pages 1269–1277, 2014.

[106] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[107] Y.-D. Kim, E. Park, S Yoo, T. Choi, L. Yang, and D. Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*, 2015.

[108] S. Lawrence, C. L. Giles, and A. C. Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1998.

[109] R. Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

[110] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[111] J. Tompson, R. Goroshin, J. Jain, Y. LeCun, and B. Bregler. Efficient object localization using convolutional networks. *CoRR*, abs/1411.4280, 2014.

[112] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[113] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

Appendix A: Proof of Results in Sec. II

Here, we prove that for the loss function defined in Eq. (3), a stationary point for N_h hidden nodes is also a stationary point of some index for N_h+1 hidden nodes if all the weights involving the additional node are zero.

Let $\mathbf{W}^{N_h}$ be a stationary point for the network with N_h hidden nodes, and $\mathbf{W}^{N_h+1}$ be the vector of weights for N_h+1 hidden nodes containing the same weights as $\mathbf{W}^{N_h}$ augmented by zero entries for the additional node. For this choice we see that

$$\begin{aligned} & y_i(\mathbf{W}^{N_h+1};\mathbf{X}) = \\ & \omega_i^{\mathrm{bo}} + \sum_{j=1}^{N_h+1} \omega_{ij}^{(1)} \tanh\left(\omega_j^{\mathrm{bh}} + \sum_{k=1}^{2} \omega_{jk}^{(2)} x_k\right) \\ &= y_i(\mathbf{W}^{N_h};\mathbf{X}) + \omega_{iN_h+1}^{(1)} \tanh\left(\omega_{N_h+1}^{\mathrm{bh}} + \sum_{k=1}^{2} \omega_{N_h+1k}^{(2)} x_k\right) \\ &= y_i(\mathbf{W}^{N_h};\mathbf{X}). \end{aligned} \tag{A1}$$

Hence $p_c(\mathbf{W}^{N_h+1};\mathbf{X}) = p_c(\mathbf{W}^{N_h};\mathbf{X})$ and $E(\mathbf{W}^{N_h+1};\mathbf{X}) = E(\mathbf{W}^{N_h};\mathbf{X})$ as well,

since $\lambda|\mathbf{W}^{N_h+1}|^2 = \lambda|\mathbf{W}^{N_h}|^2$ under these conditions. Similar results follow for the first derivatives, where

$$\begin{aligned} & \frac{\partial E(\mathbf{W};\mathbf{X})}{\partial \mathbf{W}} = -\frac{1}{|\mathbf{X}|}\sum_{d=1}^{|\mathbf{X}|} \frac{1}{p_{c(d)}(\mathbf{W};\mathbf{X})} \frac{\partial p_{c(d)}(\mathbf{W};\mathbf{X})}{\partial \mathbf{W}} \\ & + 2\lambda\mathbf{W}, \end{aligned} \tag{A2}$$

$$\text{with} \quad \frac{\partial p_{c(d)}(\mathbf{W};\mathbf{X})}{\partial \mathbf{W}} = p_{c(d)}(\mathbf{W};\mathbf{X}) \times \left(\frac{\partial y_{c(d)}(\mathbf{W};\mathbf{X})}{\partial \mathbf{W}} - \sum_{k=1}^{2} p_k(\mathbf{W};\mathbf{X}) \frac{\partial y_k(\mathbf{W};\mathbf{X})}{\partial \mathbf{W}}\right).$$

For $\mathbf{W}^{N_h}$ a stationary point of the network with N_h

hidden nodes we have

$$\frac{\partial y_i(\mathbf{W}^{N_h+1};\mathbf{X})}{\partial w_{jk}^{(2)}} = w_{ij}^{(1)} x_k \mathrm{sech}^2\left(w_j^{\mathrm{bh}} + \sum_{k=1}^{2} w_{jk}^{(2)} x_k\right)$$
$$= \begin{cases} \dfrac{\partial y_i(\mathbf{W}^{N_h};\mathbf{X})}{\partial w_{jk}^{(2)}} = 0 & 1 \le j \le N_h, \\ 0, & j = N_h + 1. \end{cases}$$
$$\frac{\partial y_i(\mathbf{W}^{N_h+1};\mathbf{X})}{\partial w_{mj}^{(1)}} = \delta_{im} \tanh\left(w_j^{\mathrm{bh}} + \sum_{k=1}^{2} w_{jk}^{(2)} x_k\right)$$
$$= \begin{cases} \dfrac{\partial y_i(\mathbf{W}^{N_h};\mathbf{X})}{\partial w_{mj}^{(1)}} = 0 & 1 \le j \le N_h, \\ 0, & j = N_h + 1. \end{cases} \tag{A3}$$

Similary for the bias weights

$$\frac{\partial y_i(\mathbf{W}^{N_h+1};\mathbf{X})}{\partial w_{j}^{\mathrm{bh}}} = w_{ij}^{(1)} \mathrm{sech}^2\left(w_j^{\mathrm{bh}} + \sum_{k=1}^{2} w_{jk}^{(2)} x_k\right)$$
$$= \begin{cases} \dfrac{\partial y_i(\mathbf{W}^{N_h};\mathbf{X})}{\partial w_{j}^{\mathrm{bh}}} = 0 & 1 \le j \le N_h, \\ 0, & j = N_h + 1. \end{cases}$$
$$\frac{\partial y_i(\mathbf{W}^{N_h+1};\mathbf{X})}{\partial w_{m}^{\mathrm{bo}}} = \delta_{im} = \frac{\partial y_i(\mathbf{W}^{N_h};\mathbf{X})}{\partial w_{m}^{\mathrm{bo}}} = 0. \tag{A4}$$

It is straightforward to show that the first derivatives also vanish for the regularisation term at the corresponding stationary points.

FIGURES

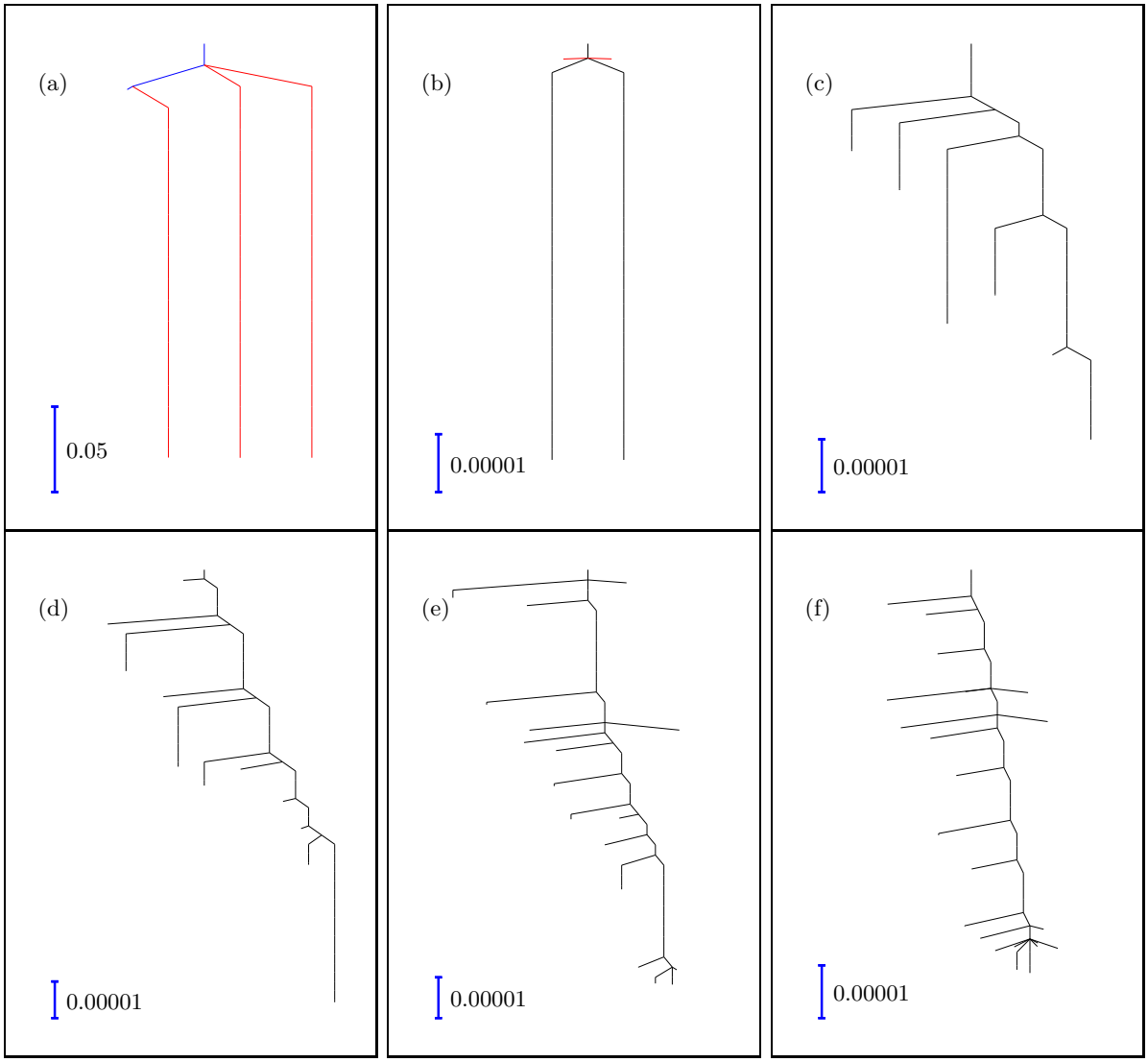

Figure 1. Disconnectivity graphs for $\lambda = 10^{-6}$ obtained with neural networks fitted to the XOR function containing N_h = (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, and (f) 6 hidden nodes. The blue bar illustrates the scale of the vertical axis that represents the energy (loss function) values. Note the contraction in scale by a factor of 5000 from 2 to 3 hidden nodes. The branch coloured blue in panel (a) corresponds to a minimum with AUC 0.5; the branches coloured red in panels (a) and (b) correspond to minima with AUC values of 0.75. All other minima have AUC values that are practically unity.

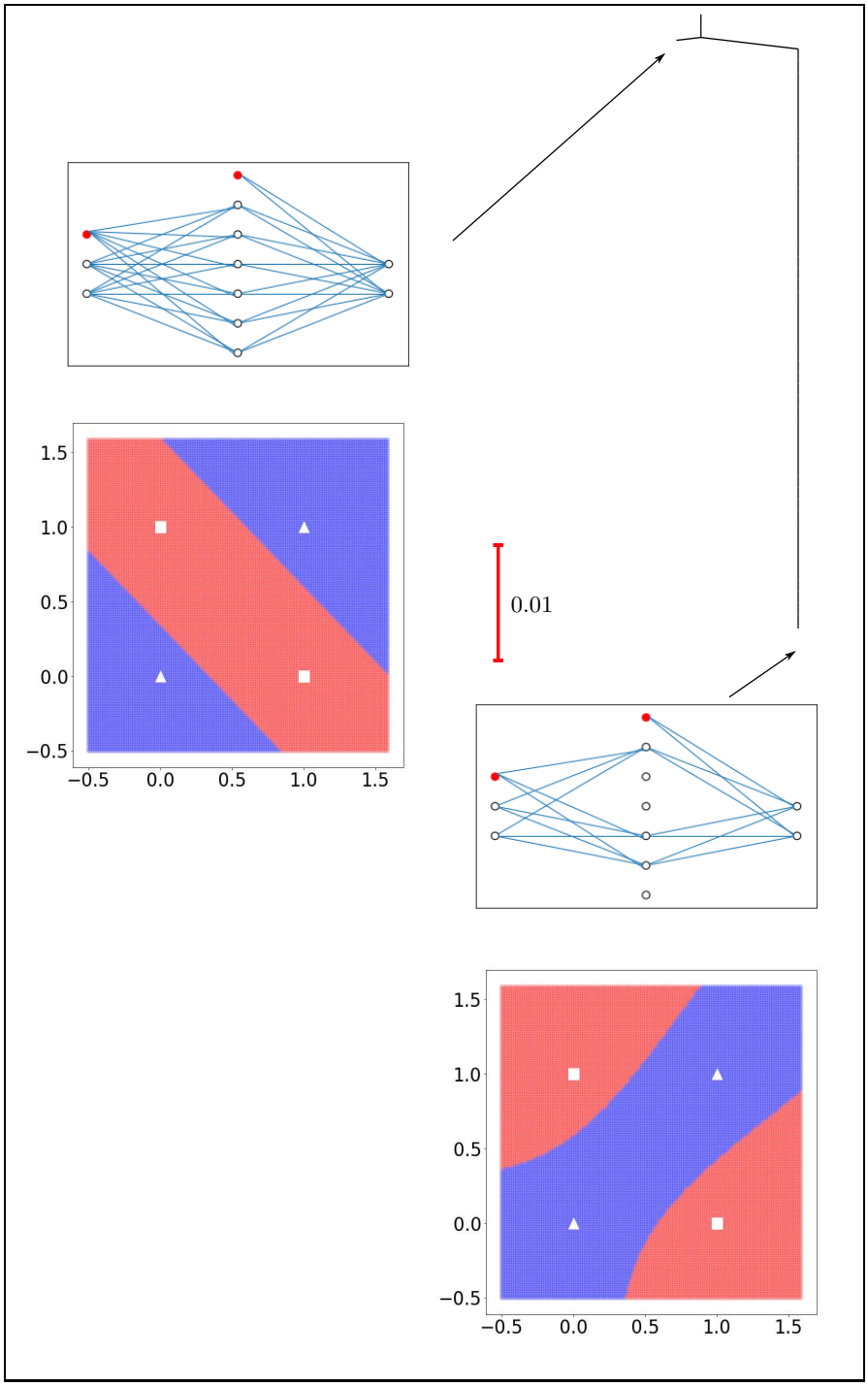

Figure 2. Disconnectivity graph for the neural network containing 6 hidden nodes with regularisation parameter $\lambda = 10^{-2}$. The figure also shows the neural network structures for the minima. The red colored nodes in the neural network diagrams are bias nodes. They are coloured red to distinguish them from the regular neurons. An edge between two nodes corresponds to non-zero weights, whereas no edge means a numerically zero weight. The figures below each network visualization illustrate the effect of perturbation to the inputs on the network output. The color of the data point with (x, y) coordinates corresponds to the output of the network for input values of x and y: red and blue points correspond to outputs 1 and 0, respectively. The white triangle and square symbols represent the 0 and 1 output, respectively, for the actual inputs present in the training set.

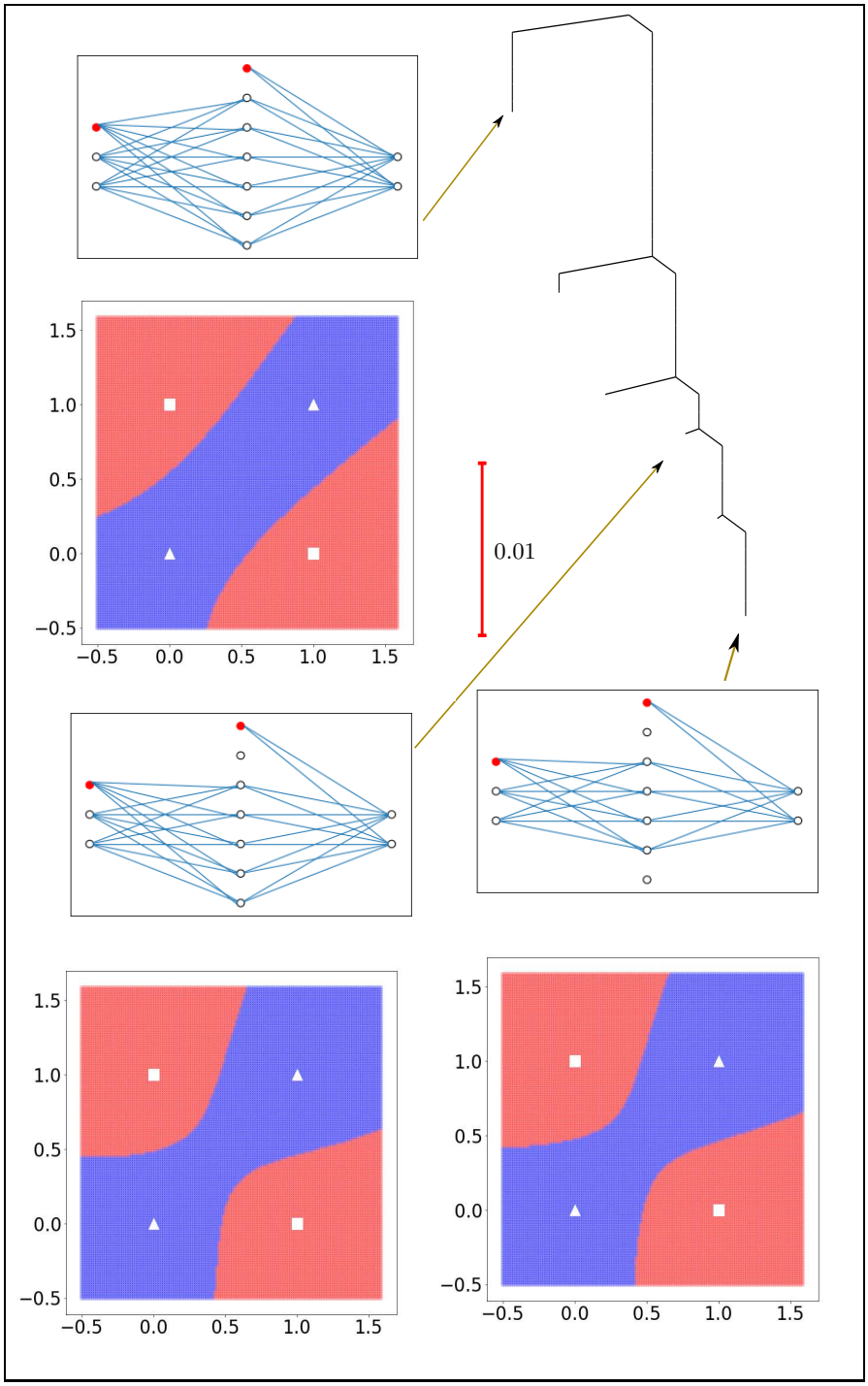

Figure 3. Disconnectivity graph for $N_h = 6$ and $\lambda = 10^{-3}$. The layout of this figure is the same as for Fig. 2.

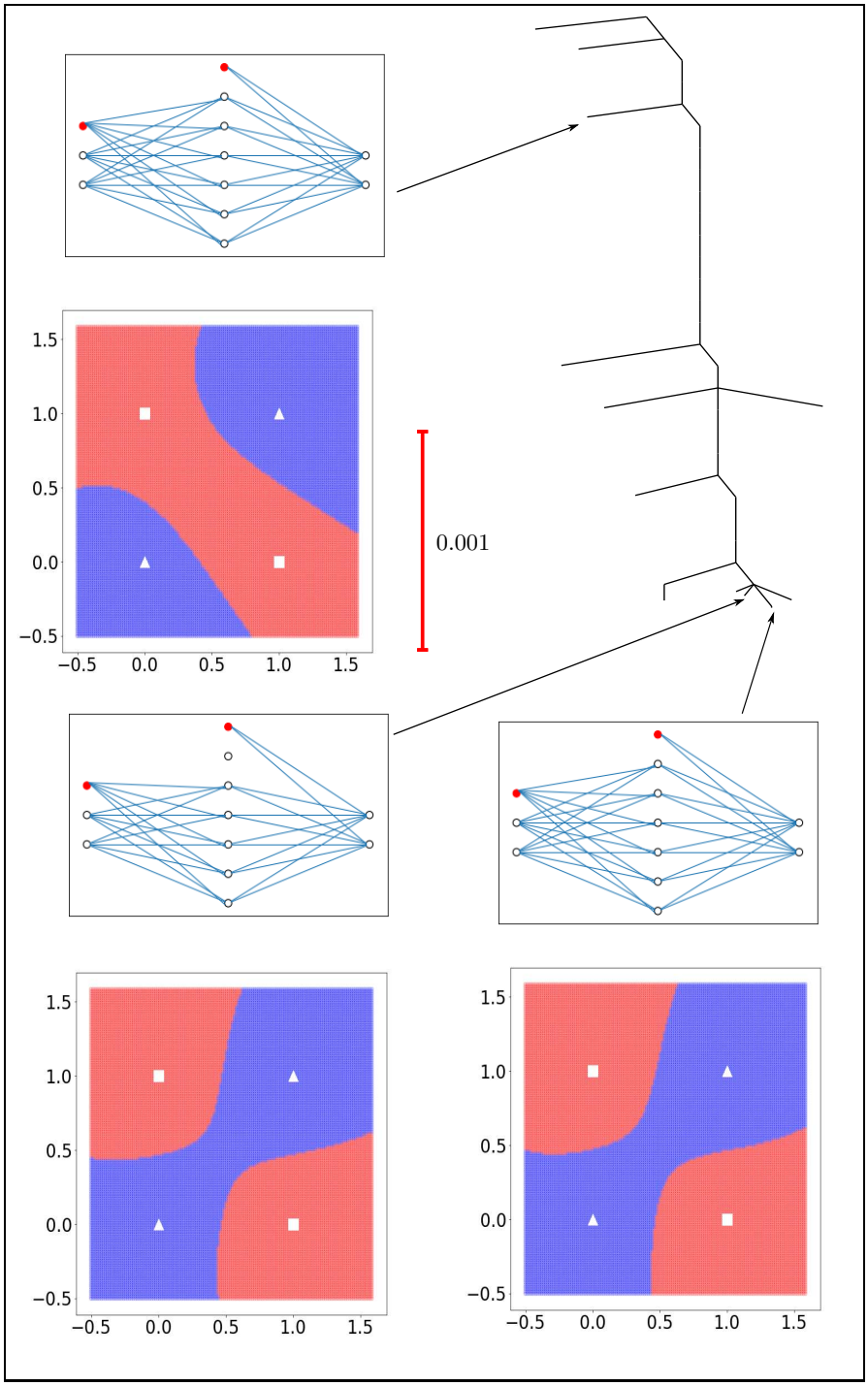

Figure 4. Disconnectivity graph for $N_h = 6$ and $\lambda = 10^{-4}$. The layout of this figure is the same as for Fig. 2.

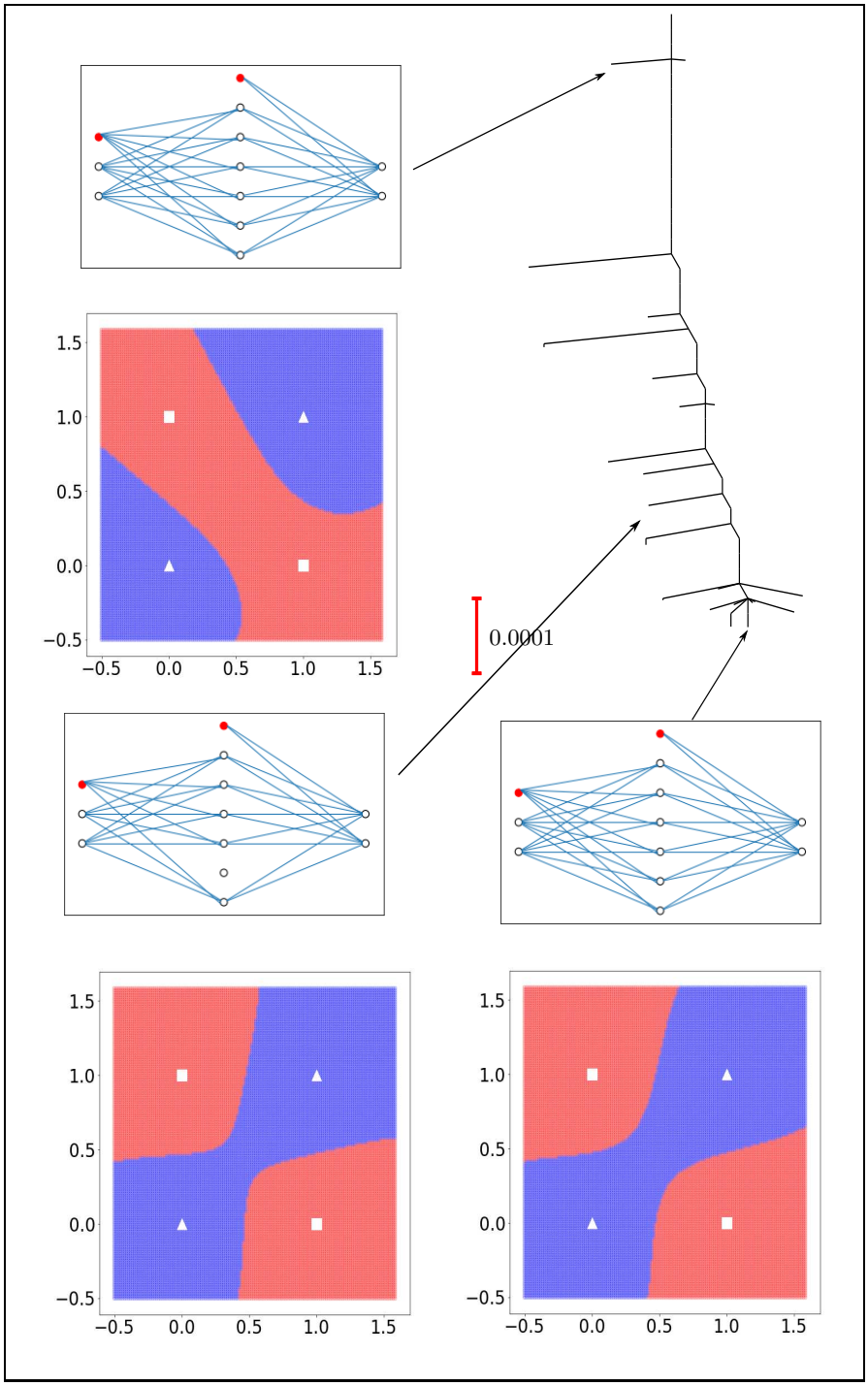

Figure 5. Disconnectivity graph for $N_h = 6$ and $\lambda = 10^{-5}$. The layout of this figure is the same as for of Fig. 2.

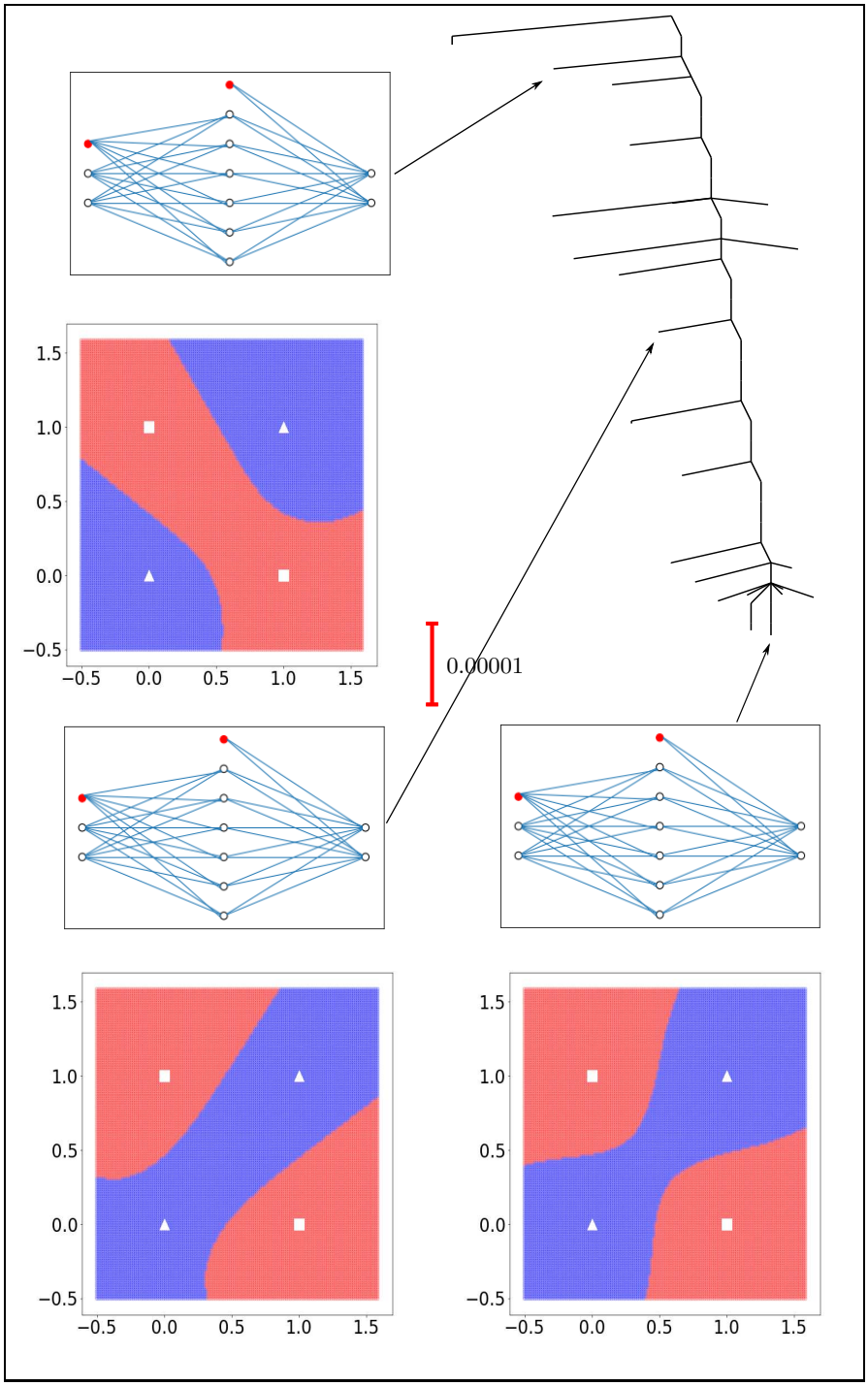

Figure 6. Disconnectivity graph for $N_h = 6$ and $\lambda = 10^{-6}$. The layout of this figure is the same as for Fig. 2.